3.¿En que consiste y cual es el funcionamiento principal de la libreria rapids?

RAPIDS es un ecosistema de código abierto desarrollado por NVIDIA para acelerar tareas de ciencia de datos y machine learning utilizando procesamiento paralelo en GPU mediante la tecnología CUDA, reduciendo drásticamente los tiempos de cómputo.

Principales Componentes:

cuDF: reemplazo GPU de pandas para manipulación y análisis de datos tabulares.

cuML: reemplazo GPU de sckit-learn que incluye modelos de regresión, clasificación, clustering, etc.

cuGraph: para análisis de grafos acelerados en GPU.

Funcionamiento:

RAPIDS convierte las operaciones de alto nivel (como .fit(), predict(), merge()) en instrucciones CUDA que se ejecutan en paralelo sobre miles de núcleos de la GPU.

Comparación con Modelos:

Regresor	Hiperparámetros	Equivalente en RAPIDS.
Linear Regressor	fit_intercept, normalize	cuml.Linear Regressor().
Lasso	alpha, max_iter, tol	cuml. Lasso().
elastic Net	alpha, L1_ratio, max_iter	cuml. Elastic Net().
Kernel Ridge	alpha, kernel, gamma	No implementado directamente, se puede emular con cuml. Kernel.
SGD Regressor	loss, penalty, alpha, learning_rate	cuml. SGD().
Bayesian Ridge	alpha_1, alpha-2, lamda-1, lamba_2	Se puede emular con cuml. Linear Regression + regularización.

Regresor	Hiperparámetros	Equivalente en RAPIDS
Gaussian Process Regressor	Kernel, alpha optimizer	No está implementado, se puede usar GPyTorch.
Support Vector Machines Regressor	C, epsilon, kernel, gamma	cuml. SVR ().
Random Forest Regressor	n_estimators, learning_rate, max_depth, subsample	cuml. ensemble. RandomForest Regressor ().
XGBoost	n_estimators, learning_rate, max_depth, colsample_bytree	xgboost. XGBRegressor (tree_method = gpu_hist) o cuml. dask. experimental. xgboost.